Mediating Multi-Party Negotiation Through Marker-Based Tracking of Mobile Phones

Michael Rohs
Deutsche Telekom Laboratories
TU Berlin, Germany
michael.rohs@telekom.de

Christian Kray
Informatics Research Institute
Newcastle University, UK
c.kray@ncl.ac.uk

ABSTRACT

Negotiating a date and a time for a meeting involving a number of people can be a difficult and time-consuming process – even when all participants are collocated and supported by technology. Oftentimes, it involves an auction-like procedure, where suggestions and conflicts are announced to the group and then checked individually until a feasible time can be found. We propose to use spatial proximity regions around handheld devices to significantly reduce the effort of exploring proposed meeting times in the context of a party of collocated people. In order to determine the location of devices on a table, we have developed a new tracking mechanism that relies on dynamic visual markers shown on the screen of the devices used. A preliminary evaluation of the underlying idea and the tracking mechanisms highlights advantages and drawbacks of our approach.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*input devices and strategies, interaction styles*; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*computer-supported cooperative work, synchronous interaction*

General Terms

Design, Human Factors

Keywords

mobile phones, marker-based tracking, meeting support, situated interaction, tangible interaction, spatially-aware interfaces, spatial relations

1. INTRODUCTION

Face-to-face meetings with business partners or with colleagues within a company constitute an important part of everyday activities in the professional lives of many people. Consequently, people frequently need to agree on a date and a time when they all will be available to meet. This task can become quite complex if there are a number of people involved or if the participants are very busy, and it may require several 'rounds' of negotiation before a date can be agreed on. A range of devices is available to support users when engaging in this process, e. g. laptop computers, PDAs, and mobile phones. Automatic solutions to the meeting negotiation problem have only been partially successful so far [5], since users often enter low-priority events or block times in their calendars that can be used for meetings if needed. Moreover, many users prefer to stay in control of their time planning rather than delegating it to an automatic tool.

In the context of this paper, we assume that all participants maintain a personal calendar using a smartphone or a PDA, and that the negotiation takes place in a location with one or more cameras for tracking phones on a table (e. g. a reasonably equipped meeting room). The task that we aim to support is the negotiation part of finding a date and a time for a meeting, in particular suggesting a date and time to the group, 'jumping' to a suggested date and time in one's own calendar, and accepting/rejecting a suggested date and time. To achieve this we use simple spatial gestures and proximity regions around handheld devices. The spatial relationships of the involved devices reflect the negotiation state for a proposed date. The location of a device on the table is determined via a new tracking mechanism that relies on dynamic visual markers shown on the device screen.

The benefits of fine-grained spatial interaction for handheld devices have been recognized some time ago [2], but the idea has been difficult to implement. *TRIP* [1] is a vision-based location system using circular markers. The markers are tracked by cameras installed in the environment. It would be possible to show TRIP tags on phone displays, but their circular form factor would lead to a lot of wasted display space and prevent simultaneous display of application data. *Relate* [3, 4] is a system for the detection of spatial relationships between collocated mobile devices. It does not require an infrastructure, but determines relative position and orientation via ultrasound sensing implemented on USB dongles. This approach requires additional hardware on each device, whereas the marker tracking approach presented here is a software-only solution on the users' side. Our approach requires an external infrastructure, but has a higher accuracy and update rate. Doodle (www.doodle.ch) is an online tool for meeting negotiation. The initiator proposes a number of dates and sends a unique URL generated by the system to all participants, who can then individually accept or decline the proposed dates. This corresponds to one round of negotiation in our system, albeit with multiple proposed dates.

2. APPROACH

In this section we describe our approach to achieve the goals outlined above. We first present and motivate the basic idea before giving a detailed description of how we realized it technically. We also conducted a preliminary evaluation of both aspects, which we discuss in section 3.

2.1 Basic Concept

The basic idea underlying our approach is to use spatial proximity regions around mobile phones to trigger particular actions in response to other devices entering them, leaving them, or staying within them for a certain amount of time. For the scenario at hand, we define three regions around a mobile phone, and assign them to synchronization-related actions in the context of the meeting negotiation process. Figure 1 depicts a diagram of the different regions around a mobile device.

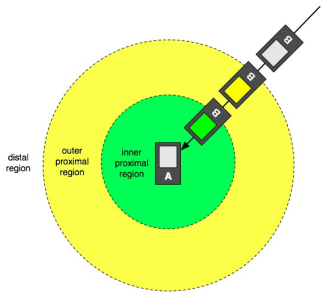

Figure 1: Proximity regions around a stationary mobile phone; see text for explanation.

We call the region furthest from a device its *distal region (DR)*. As long as devices are located in this region relative to a device, they are considered to be too far away for interaction and do operate independently. In this state, it is hence possible for the respective owners of these devices to individually explore their personal calendar.

The interaction is initiated once a device B enters the second region around a device A, which we call its *outer proximal region (OPR)*. In this case, we assume that the owner of the mobile phone A has selected a particular date and time and is proposing it to the group as a potential meeting time. The other members of the group can then check whether they are available at that time by moving their own devices close to the device of A. Once a device B enters the OPR of device A, the calendar application on device B automatically jumps to the date and time that is selected on device A. The owner of device B can immediately see whether the meeting time suggested by A is still available in their own calendar. By moving device B out of the OPR of A and into its DR, the calendar on device B reverts back to the state it was in before entering the OPR of A

In order to confirm a suggested meeting time, a device needs to enter into and remain within the region nearest to the device proposing the date and time; we call this region its *inner proximal region (IPR)*. As shown in Figure 1, once device B enters the IPR of device A, it confirms the suggested meeting time. User B can abort the confirmation by moving device B out of the IPR of A. If there are more than two parties involved, *all* devices need to be in the IPR of A in order to confirm the suggested date and time. As long as there are still participating devices outside the IPR of A, those inside the IPR will merely indicate that they are ready to commit to the date and time but will not accept it. Figure 2 illustrates this behavior: in case (a), device C is still outside the IPR of A. Therefore, the suggested date is not yet accepted. In case (b), both C and B are inside the IPR of A; hence, the suggested meeting date is accepted by all participants.

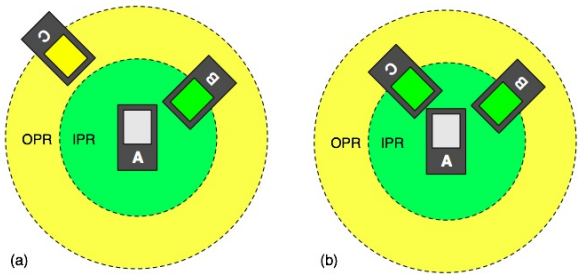

Figure 2: Multi-party negotiations: (a) no commitment as not all devices are inside IPR of A, (b) commitment performed as all devices are inside IPR.

The owner of device A can abort the interaction at any stage by moving device A in any direction. (Covering the visual marker on the display has the same effect as it effectively removes the device from the system.) This also invalidates the suggested time/date and all other devices within the proximal regions of device A revert back to the state they were in prior to entering into different stages of commitment. Table 2 summarizes the different actions and the corresponding negotiation stages.

Table 1: Spatial actions involving devices A and B and the negotiation stages associated with them.

Spatial action	Negotiation stage
B in DR of A	no interaction, i. e. independent operation
B in OPR of A	B displays date/time selected on A (exploration)
B in IPR of A	B accepts date/time selected on A (commitment) – also see text
moving B into DR of A	interrupts interaction
moving A while device are in its OPR/IPR	interrupts interaction

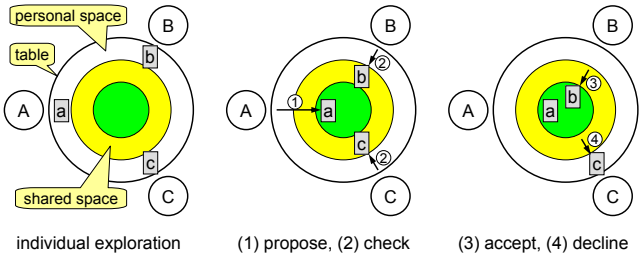

Figure 3: Users A, B, C at a table: (left) individual exploration of personal calendars; (middle) A proposes a date, B and C check the date; (right) B accepts and C declines the proposed date.

An alternative approach to the one described above is to define regions with respect to the physical position of the persons involved. This helps to increase scalability to larger tables and groups, since users do not have to move their devices over a long distance towards to the proposing device.

As shown in Figure 3 the area close to a user is defined as a personal space, in which one's own calendar can be explored. By moving a phone into the central region, a user can propose a meeting time. Other participants can explore the suggested time by moving their devices into the shared space, accept it by moving them into the central area, or reject it by moving their device back into the personal space.

2.2 Technical Realization

The position and orientation of the phones is tracked by a dynamic visual marker that is displayed on the phone screen (see Figure 4). The marker is designed in such a way that it occupies only a small amount of screen space. In the prototype implementation on a Nokia N95 with a screen size of 240×320 pixels the marker just occupies 15% of the screen area. The marker is placed at the top of the screen in order to make it unlikely that users inadvertently cover the marker with their hands. The phone marker stores 42 bits of data that are protected by a (48,42,3) linear code with a Hamming distance of 3. For each detected marker the recognition algorithm provides the encoded value, center position, rotation, and distance.

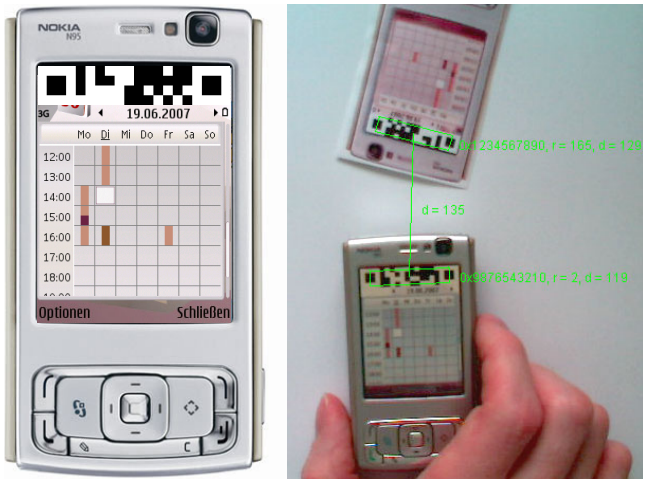

Figure 4: Visual marker at the top of a phone calendar application (left). Tracking of two phones (right). Marker values and distance overlaid.

Figure 5 shows the layout of a marker. It consists of two square corner stones surrounded by whitespace and a central data area of 12×4 elements. The total horizontal width of the marker is 20 elements. The recognition algorithm proceeds as follows: The grayscale image is thresholded to a black-and-white image. To find the corner stones connected regions are computed. Regions with an axis ratio greater than 0.7 (minor to major axis length) and a pixel count within a certain range are classified as corner stone candidates. Then, matching pairs of corner stones are identified. Ideally, matching corner stones have the same size and a distance of 16 elements. Since markers can appear tilted in the camera image, we allow a ±25% deviation from the ideal values of size and distance. In order to sample the data points of a potentially tilted marker, a homography is computed based on four points near the corner stones (indicated by diamond shapes in Figure 5). These points are found by moving ±1.5 elements in a direction perpendicular to the

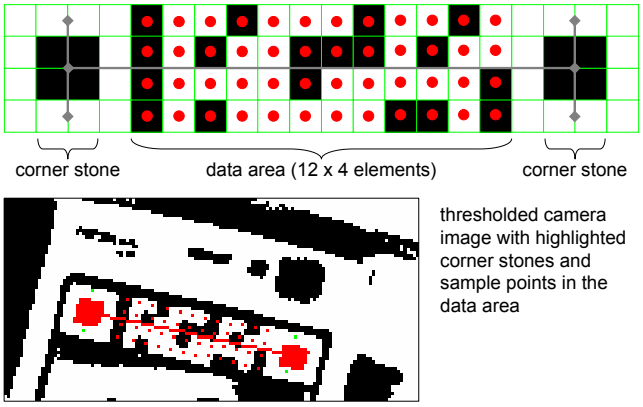

Figure 5: Elements of the visual marker for display on phone screens (top) and thresholded camera image with highlighted structures (bottom).

connection vector of the two corner stones. The sampled 48 points in the data area then undergo a parity check. The resulting 42 data bits are reported together with the position of the marker (within the camera coordinate system), its rotation, and its distance (inverse of size multiplied by a constant).

Markers are tracked by a downward-facing camera mounted above the table. We used a typical webcam (Logitech QuickCam Fusion) with a wide angle lens (diagonal angle of view about 70°). The wide viewing angle means that the camera cannot be ceiling-mounted but at distances between 30-50 cm only. For ceiling-mounted operation a telephoto lens would be needed. The tracking software is implemented in Java. Initially we connected the video stream via the Java Media Framework (JMF). While being platform independent, the drawbacks of this solution were the long delay (about 500 ms) and the limited resolution (maximum 640×480 pixels). We therefore switched to Windows DirectShow and Java Native Interface (JNI). The delay is how shorter and higher resolutions can be selected (up to 1280×960 pixels). Marker sightings are reported via Bluetooth to the phones connected to the PC. In the prototype implementation each phone computes its distance to the closest marker, determines in what proximity region it is and updates its display accordingly.

3. PRELIMINARY EVALUATION

In order to evaluate our approach, we performed a qualitative analysis of the effort involved in negotiating meeting comparing our approach to a base case. In addition, we compiled an initial characterization of the marker-based visual tracking mechanism. At this stage, both are preliminary but nevertheless provide initial support for our approach. We are also planning a study involving human subjects at a later stage.

3.1 Qualitative Analysis of Efficiency

As described in section 1, our aim is to improve the negotiation of a meeting date among multiple people. We assume that they use a mobile device (such as a mobile phone) to maintain their personal calendar. We used the standard calendar application of a high-end mobile phone (Nokia N95)

as a base case.[1] Figure 6 depicts a simplified state diagram of the calendar application. In our analysis, we disregarded any shortcuts that were not accessible/visible through the user interface displayed on the screen. In the figure, arrows correspond to transitions between states, and the numbers correspond to the number of key presses that are needed to get from one state to the other. For example, in order to bring up the form for creating a new meeting on the day selected in month view, four consecutive key presses are needed. There are two steps in particular, for which we want to reduce the effort: exploring a date suggested by another person in one's own calendar, as well as confirming a suggested meeting date and entering it into one's own calendar.

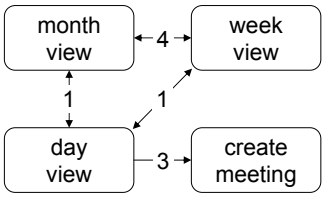

Figure 6: Simplified state diagram of the calendar application on a mobile phone (Nokia N95); arrows are labeled with the number of key presses needed to get from one state to the other.

In order to estimate the time needed to complete these tasks in the base case, we first need to determine the number of key presses a user has to perform during this process. This number depends on several factors:

- the view (month, week, day) that is currently selected
- the temporal distance between the currently selected day and the suggested date
- the number of meetings that are already scheduled for the suggested date.

Both month and week view are organized as 2D grids, which the user can navigate using the four-directional navigation button. The day view is a linear list, which also uses the navigation button. We assume that only the day view enables the user to fully assess whether a time slot is available for a meeting or not. (The week view offers only a coarse view of time allocation but does not provide details about scheduled meetings such as their exact starting and end times.) Based on these considerations we can estimate the number of key presses needed to navigate to a suggested date (see Table 2).

It should be noted that Table 2 only represents an estimate, and that there are additional key presses needed if the day suggested already has a number of meetings scheduled. The minimum number of key presses can only be realized if $t_c = t_s$; the further apart the dates are, the more key presses are needed. Assuming optimal usage, there is an upper bound: by entering the pop-up menu and selecting the 'go to date' option, users can enter a target date numerically. The cost incurred by this operation is at least 9 key presses (11 if a different month is suggested and 15 if the suggested date is in another year).

[1]We briefly explored the calendar tool that comes preinstalled with a mid-range phone (Sony Ericsson K750i) and found it to be very similar to the one on the N95.

Table 2: Key presses (KP) needed to navigate from current date t_c to a suggested date t_s.

view	key presses needed	min
month	1 KP per every week t_c and t_s are apart 1 KP per every day in the week t_c and t_s are apart 1 KP to enter day view	1 KP
week	7 KP per every full week t_c and t_s are apart 1 KP per every day in the week t_c and t_s are apart 1 KP to enter day view	1 KP
day	1 KP per every day t_c and t_s are apart	0 KP

In order to illustrate the key presses needed to navigate to a suggested date, let us consider the following example. A user is in month view and has selected Friday, 22 June. Another person suggests Thursday, 28 June as a meeting date. In this case, three key presses are required. In week view seven key presses would be required, and in day view six. If the suggested date would have been 12 July instead, five key presses would be required in month view, 21 in week view and 20 in day view. In each view, additional key presses would be required in case the suggested day already has more meetings scheduled than can shown at once.

This clearly demonstrates that under all but optimal conditions (i. e. the suggested date is the one already selected) it is reasonable to assume that a user will have to perform between 3 and 9 key presses, oftentimes more. In addition, the original state will be lost. Returning to the date that was originally selected incurs the same number of key presses again. Consequently, the costs will be doubled for every suggested date that is explored but then rejected.

If a date is accepted, further key presses are required. Depending on the view the user is in at that time, three or four key presses are needed to enter the form to create a meeting. In this form, a user has to fill in a number of fields (subject, location, start time, end time, etc.). Minimally, a start and end time have to be provided, which results in at least four key presses (if the automatically suggested times exactly match the proposed meeting times), respectively 14 individual key presses (if start and end time have to be entered manually).

While it is difficult to compare key presses to gestures (i. e. moving a device on a table), Table 3 illustrates that the proposed approach is at least less complex for the user: a single action is required to navigate to a suggested date, to reject the suggestion and return to the previously selected date, and to accept a suggested date. In the base case scenario, a user will also have to navigate through a number of forms or menus, which adds to the complexity, whereas this is not the case in our approach. Furthermore, if the user wants to include additional information about the meeting (e. g. its subject and location), a significant number of key presses will be required. Using our approach, if the proposer included this information in the suggestion, this would not incur any additional costs for other participants.

Table 3: Complexity of interaction to perform a single step of negotiation

task	base case	our approach
exploration	ca. 3 to 9 key presses	1 gesture
rejection	ca. 3 to 9 key presses	1 gesture
acceptance	ca. 4 to 14 key presses	1 gesture

3.2 Characterization of Tracking Mechanism

Our experiments show that the minimum size of a marker element is about 3×3 pixels in the camera image. This means that the minimum required marker width is 20×3 = 60 pixels. Since the display width of our prototype device is 40 mm, the table surface must be sampled at a minimum resolution of 60/40 = 1.5 pixels/mm. At a camera resolution of 1280×960 pixels a table area of 85×64 cm can thus be covered by a single camera. For robustness a slightly smaller physical area may be desirable. On a notebook with a 1.73 GHz Intel Pentium M and 1 GB of main memory the average processing time of the algorithm for a 1280×960 image is 115 ms, 37% of which is due to thresholding. However, JNI and other components introduce additional delays.

Since the marker's corner stones are symmetrical their orientation has to be checked by other means. One possibility would be to reserve two data elements as orientation indicators. Another way, which is currently implemented, is to use the error check as an implicit orientation indication. If the error check fails in one direction, the sampled bits are reversed and checked again. Given that the error protecting code is not symmetric only one direction will succeed.

4. DISCUSSION

After presenting an initial evaluation of the proposed approach, we want to discuss a number of questions relating to the performance, feasibility and other characteristics of using visual markers in this way.

Does the approach improve performance? As highlighted in the previous section and Table 3 further experiments are needed to more precisely characterize the performance of our approach. It is however fairly clear that using our approach significantly simplifies the whole process by reducing the complexity of the interaction needed. In particular, the cost of exploring suggested dates is reduced.

Is the approach suitable for meeting negotiation? Assuming that the date and time of a meeting is oftentimes agreed at the end of another meeting, we would argue that it is reasonably realistic to assume a setting where all participants are collocated and are using personal devices to access their individual diaries. A typical meeting room usually provides a technical infrastructure, which might well include a camera to record a meeting (e. g. the slides presented during the meeting) or equipment for video-conferencing. Even if this not the case, the cost of adding a simple webcam are very low. Based on the argument, outfitting more casual places (such as a café in a business district of a big city) would not be prohibitively expensive.

How to deal with privacy issues? Obviously, camera-based tracking can pose a threat to the privacy of participants, i. e. the camera can record the whole screen content, potentially exposing details from individual diaries. There are three different responses to this:

1. We assume a benevolent environment (e. g. a meeting room owned by a trusted party).
2. We improve the overall security of the system. Examples for this approach include only showing markers on the screen and limiting the infrastructure to tracking the location of the markers.
3. We eliminate the need for an infrastructure. An idea we plan to investigate would be to use the camera on the mobile phone of a participant.

5. SUMMARY

In this paper, we presented a novel approach to negotiate meeting times among a group of collocated participants. In order to realise it, we used proximity regions and a tracking mechanism based on a new type of visual markers that are being displayed on the screen of handheld devices such as mobile phones or PDAs. We provided a preliminary evaluation of the proposed approach, both with respect to its interaction properties and to the characteristics of the tracking scheme. Based on these results, we discussed the benefits in terms of its performance and feasibility for the example application. We also reviewed aspects related to privacy, and we outlined three approaches to increase the 'security' of our approach.

Due to the overall promising results, we intend to further develop the approach. One interesting direction is to use the camera of a participant's device instead of one provided by the environment – thereby making the system independent of any external infrastructure. Other aspects we want to investigate are the use of dynamic markers as a visual communication channel and the use of the system for other applications.

6. ACKNOWLEDGMENTS

We would like to acknowledge a hardware grant by Nokia Research Labs, Finland that provided the N95 handsets used. We would also like to thank the organizers and participants of the FTIR workshop in Münster, Germany, for inspiring discussions.

7. REFERENCES

[1] D. L. de Ipiña, P. R. S. Mendonça, and A. Hopper. TRIP: A low-cost vision-based location system for ubiquitous computing. *Personal Ubiquitous Comput.*, 6(3):206–219, 2002.

[2] G. W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Commun. ACM*, 36(7):39–49, 1993.

[3] M. Hazas, C. Kray, H. Gellersen, H. Agbota, G. Kortuem, and A. Krohn. A relative positioning system for spatial awareness of co-located mobile devices and users. In *Proceedings of MobiSys 2005*, pages 177–190, 2005.

[4] G. Kortuem, C. Kray, and H. Gellersen. Sensing and visualizing spatial relations of mobile devices. In *Proceedings of UIST 2005*, pages 93–102, 2005.

[5] L. Palen. Social, individual and technological issues for groupware calendar systems. In *Proceedings CHI'99*, pages 17–24, 1999.